Department of Veterans Affairs
Mental Health – Suicide Prevention
Suicide Prevention Package

May 2021

Version 1.2

Technical Manual

YS*5.01*158

Submitted as CLIN 0004AV

Contract VA118-16-D-1007, Task Order VA11817F10070006

Submitted by:

Booz Allen Hamilton Inc.

141 W. Front Street, Suite 200

Red Bank, NJ 07701

Phone: 732-936-3500

Fax: 732-936-3535

Booz | Allen | Hamilton

Revision History

Date	Revision	Description	Author
02/28/2021	1.2	Update – Build 14 • Update application name from MHA PaSE to MHA Web	Booz Allen
12/11/2020	1.1	Update – Build 13	Booz Allen
09/11/2020	1.0	Initial Document	Booz Allen

Table of Contents

1. Introduction 1
   1.1. Purpose 1
   1.2. System Overview 1
   1.3. Document Orientation 1
2. Implementation and Maintenance 2
   2.1. System Requirements 2
   2.2. System Setup and Configuration 3
3. Files 3
4. Routines 3
5. Exported Options 3
6. Mail Groups, Alerts, and Bulletins 3
7. Public Interfaces 3
   7.1. Integration Control Registrations 3
   7.2. Application Programming Interfaces 3
   7.3. Remote Procedure Calls 5
   7.4. HL7 Messaging 5
   7.5. Web Services 5
8. Standards and Conventions Exemptions 5
   8.1. Internal Relationships 5
   8.2. Software-wide Variables 6
9. Security 6
   9.1. Menus and Options 6
   9.2. Security Keys and Roles 6
   9.3. File Security 6
   9.4. Electronic Signatures 6
   9.5. Secure Data Transmission 6
10. Archiving 6
11. Non-Standard Cross-References 6
   11.1. Enterprise Service Desk and Organizational Contacts 7
12. Acronyms and Abbreviations 7

1. Introduction

This document gives a brief technical overview of the Mental Health Assistant (MHA) Web-Patient Entry (MHA Web-PE) Application architecture and describes the developer environments.

1.1. Purpose

This document describes the technical details for MHA Web.

The project scope can be found in the Suicide_Prevention_Requirements.xlsx located in GitHub.

1.2. System Overview

The MHA Web application will give providers ability to assign and schedule current and future patient self-administrations that can be completed on a kiosk or iPad, comprehensively manage assessments, securely implement mobile instrument administrations and migrate data to VistA. The new enhancements which allow clinicians to assign instruments for patient self-administration require navigation from existing MHA and Computerized Patient Record System (CPRS) screens. This ensures efficient processes and that the facilitation of management of access to care for at risk Veterans.

The full capabilities of this application will be delivered in multiple iterations. The first iteration provides the ability for a clinician to create a patient Assignment in the MHA Delphi interface and allows the patient to self-administer the instrument Assignment on a kiosk or iPad at a VA clinic (MHA Web–PE). The second interation allows the clinician to create and edit a patient assignment in the web based interface and allows the clinician to administer the instrument. This is known as Planning and Staff Entry (MHA Web).

1.3. Document Orientation

1.3.1. Software Disclaimer

This software was developed at the Department of Veterans Affairs (VA) by employees and contractors of the Federal Government in the course of their official duties. Pursuant to title 17 Section 105 of the United States Code this software is not subject to copyright protection and is in the public domain. VA assumes no responsibility whatsoever for its use by other parties, and makes no guarantees, expressed or implied, about its quality, reliability, or any other characteristic. We would appreciate acknowledgement if the software is used. This software can be redistributed and/or modified freely provided that any derivative works bear some notice that they are derived from it, and any modified versions bear some notice that they have been modified.

1.3.2. References

N/A

2. Implementation and Maintenance

MHA Web is implemented in Spring Boot with a React front-end and Java backend. The React code is bundled using webpack and deployed onto the Spring Boot server running embedded Tomcat.

2.1. System Requirements

The MHA Web application requires:

1) Microsoft Azure cloud-based server environment
2) Red Hat Linux (Redhat: enterprise_linux:7.6:GA:server)
3) Java JRE (Java Open JDK 1.8.0_191)
4) Docker (18.09.1, build 4c52b90)
5) MHA Web-PE JAR file

2.1.1. Hardware Requirements

There are no hardware modifications with this Mental Health Assistant release; however, previous packages have been run on the standard hardware platforms used by the Department of Veterans Affairs Health Care System facilities. These systems consist of standard or upgraded Alpha AXP clusters, standard intel hardware Windows operating system, or Microsoft Azure instances and run either Cache-VMS, Cache-NT, Cache- OpenVMS, or Cache- Windows Server 2008 or higher.

Deployment at the VistA sites require a workstation, kiosk, or VA iPad for patients to access MHA Web-PE. A VA standard workstation is required to access MHA Web.

2.1.2. Software Requirements

Application Name	Minimum Version Needed
CPRS	31 A
Clinical Reminders	2.0
Kernel	8.0
RPC Broker	1.1
PIMS	5.3
VA FileMan	22.0
Mailman	8.0

The following patches are required:

YS*5.01*141 – MHA GUI and Web Updates

YS*5.01*150 – Suicide Prevention Instruments

YS*5.01*173 – Inactivate I9 Instruments, Update PROMIS29

The following patch is strongly recommended:

DG*5.3*1026 – Master Veteran Indexa VistA Enhancement – TFL API Update

2.1.3. Database Requirements

MHA Web requires a compatible MySQL Database hosted in the Azure environment. This database serves as connection and session management and does not contain any Protected Health Information (PHI) or Personally Identifiable Information (PII).

2.2. System Setup and Configuration

MHA Web runs in an Azure cloud environment as a JAR file in a Dockerized instance. The application instance properties and connection parameters are contained in a configuration files.

- application.properties
- application.yml

3. Files

Two files are required for the MHA Web application. The first is the executable JAR file and the second is the configuration file which contains the properties and connection parameters for the application instance.

4. Routines

There are two components to the routines, one which is the web-based UI and the other manages the VistA Remote Procedure Calls (RPC).

- staffentry.jar – Executable file which runs in the Azure Docker container.
- YS_501_158.KID – VistA KIDS package of routines which handle RPC requests.

5. Exported Options

MHA Web adds YTQREST MHA to the OPTION file as the context for the RPC call.

6. Mail Groups, Alerts, and Bulletins

There are no mail groups, alerts or bulletins.

7. Public Interfaces

There is an intranet clinician facing web site that hosts the MHA Web application. The web site is accessed by the browser on a workstation within the VA.

7.1. Integration Control Registrations

To view the details of a Database Integration Agreements (DBIA) use the VA Forum option "Inquire to an Integration Control Registration" to display the DBIAS where the Mental Health package is the custodial or the subscribing package.

7.2. Application Programming Interfaces

MHA Web utilizes two APIs.

The Azure docker React/Java application leverages the existing VistaLink APIs. Further documentation regarding the available APIs are available on the VA Software Document Library.

The VistA interface exposes application functionality through the RPC, YTQREST QADMIN. The functional calls are listed in the table below.

RPC URL	Description
GET /api/mha/getconn	Response to an RPC connection test request
GET /api/mha/patient/:dfn/identifiers	Get patient demographic information for the patient specified by the :dfn parameter
GET /api/mha/persons/:match	Get a list of users from the NEW PERSON file that starts with the :match parameter
GET /api/mha/users/Lmatch	Get a list of users from the NEW PERSON file that starts with the :match parameter
GET /api/mha/instruments/active	Get a list of active instruments
GET /api/mha/instrument/:instrumentName	Get the instrument definition specified by the :instrumentName parameter
GET /api/mha/checks/:instrumentName	Get validation requirements for the instrument specified by the :instrumentName parameter
GET /api/mha/assignment/:assignmentId	Get information regarding the patient instrument assignment specified by the :assignmentId parameter
GET /api/mha/assignment/:assignmentId/:division	Get information regarding the patient instrument assignment specified by the :assignmentId parameter
GET /api/mha/assignment/graph/:dfn/:instrument	Get the graph data for a patient specified by :dfn and instrument administrations specified by :instrument.
GET /api/mha/instrument/admin/:adminId	Get the current state and details of an instrument administration in progress specified by the :adminId parameter.
GET /api/mha/instrument/report/:adminId	Get the report text for the instrument administration specified by :adminId
GET /api/mha/instrument/note/:adminId	Get the note text for the instrument administration specified by :adminId
GET /api/mha/permission/cosign/:adminId/:userId	Get the cosigining permission for the instrument administration specified by :adminId and the user specified by :userId
GET /api/mha/instrument/list/:dfn	Get a list of instrument administrations for a patient specified by :dfn
GET /api/mha/location/list/:duz	Get a list of Hospital Locations available for a user specified by :duz
GET /api/mha/category/list	Get a list of instruments available sorted by instrument type Category

RPC URL	Description
GET /api/mha/assignment/list/:dfn	Get a list of Assignments for a patient specified by :dfn
GET /api.mha/consult/list/:dfn	Get a list of Consults for a patient specified by :dfn
GET /api/mha/assignment/staff/:assignmentId	Get a Staff only assignment specified by :assignmentId
POST /api/mha/assignment	Save a new Assignment
POST /api/mha/assignment/edit/:assignmentId	Save the edited Assignment specified by :assignmentId
POST /api/mha/instrument/admin	Save the status and details of an instrument administration in progress.
POST /api/mha/instrument/note	Save the unsigned Note text for an instrument administration
DELETE /api/mha/assignment/:assignmentID	Delete the Assignment specified by :assignmentId
DELETE /api/mha/assignment/:assignmentID/:division	Delete the Assignment specified by :assignmentId

7.3. Remote Procedure Calls

MHA Web uses one RPC, YTQREST QADMIN.

```
NAME: YTQREST QADMIN                    TAG: QADMIN
  ROUTINE: YTQREST                    RETURN VALUE TYPE: ARRAY
  AVAILABILITY: RESTRICTED             WORD WRAP ON: FALSE
  VERSION: 1                           APP PROXY ALLOWED: No
 DESCRIPTION:
 Front controller for questionnaire administrator.
```

7.4. HL7 Messaging

No HL7 interfaces are used.

7.5. Web Services

No web service calls are used.

8. Standards and Conventions Exemptions

No exemptions are used.

8.1. Internal Relationships

No internal relationships.

8.2. Software-wide Variables

No software-wide variables are used.

9. Security

Clinicians use CPRS and MHA to create Assignments for patients and administer instruments. No changes will be needed to the security and privacy requirements already approved for VistA and the GUI applications.

MHA Web functionality adheres to all VA and VHA security requirements and allows the clinician to create Assignments for patients, administer instruments, and review results.

9.1. Menus and Options

One new OPTION is created to provide the Context for the RPC, YTQREST MHA.

The OPTION YS BROKER1 is updated to include the RPC YTQREST QADMIN. This allows users who already have YS BROKER1 to access the application without any additional configuration.

9.2. Security Keys and Roles

No new Security Keys or roles are defined.

9.3. File Security

MHA Web does not store any PHI/PII. The application transmits PHI/PII to and from Vista via https and VistaLink but sensitive data is only hosted on Vista systems.

9.4. Electronic Signatures

No changes to Electronic Signatures are implemented.

9.5. Secure Data Transmission

MHA Web-PE only transmits data inside the VA network via https and VistaLink.

10. Archiving

No archiving is needed.

11. Non-Standard Cross-References

MHA Web-PE utilizes the standard format of the ^XTMP global to store Assignment information. Assignments are only stored in ^XTMP until they are completed. In order to increase application efficiency MHA Web-PE makes use of two cross-references in ^XTMP.

XTMP Cross-Reference	Description
^XTMP("YTQASMT-INDEX","AC",ssn,lastname,date/time) =assignment#	Assignment cross-reference by last four SSN, patient last name, and inverse date/time.

XTMP Cross-Reference	Description
^XTMP(“YTQASMT-INDEX”,”AD”,dfn,duz,assignment#) =date	Assignment cross-reference by patient DFN, user DUZ, and assignment#. Date is in FileMan format.

11.1. Enterprise Service Desk and Organizational Contacts

Contact the Enterprise Service Desk if assistance is needed.

12. Acronyms and Abbreviations

Term	Meaning
CPRS	Computerized Patient Record System
DSM-5	Diagnostic and Statistical Manual of Mental Disorders – 5th Edition
https	Hypertext Transfer Protocol Secure
MHA	Mental Health Assistant
MUMPS	Massachusetts General Hospital Utility Multi-Programming System
PSPO	Patient Safety Program Office
PTSD	Posttraumatic Stress Disorder
RPC Broker	Remote Procedure Call Broker
RSD	Requirements Specification Document
RTM	Requirements Traceability Matrix
SAC	Standards and Conventions
SDD	System Design Document
SSO	Single Sign-On
SPP	Suicide Prevention Project
SQA	Software Quality Assurance
TRM	Technical Reference Model
VA	Veterans Administration
VDD	Version Description Document
VistA	Veterans Health Information Systems and Technology Architecture